Trinity Advanced Power Management

November 14, 2018

SC'18 BOF:
A Look Ahead – Energy and Power Aware Job Scheduling and Resource Management

Kevin Pedretti, ktpedre@sandia.gov

Sandia National Laboratories

Exceptional service in the national interest

U.S. DEPARTMENT OF ENERGY

NNSA National Nuclear Security Administration

CCR Center for Computing Research

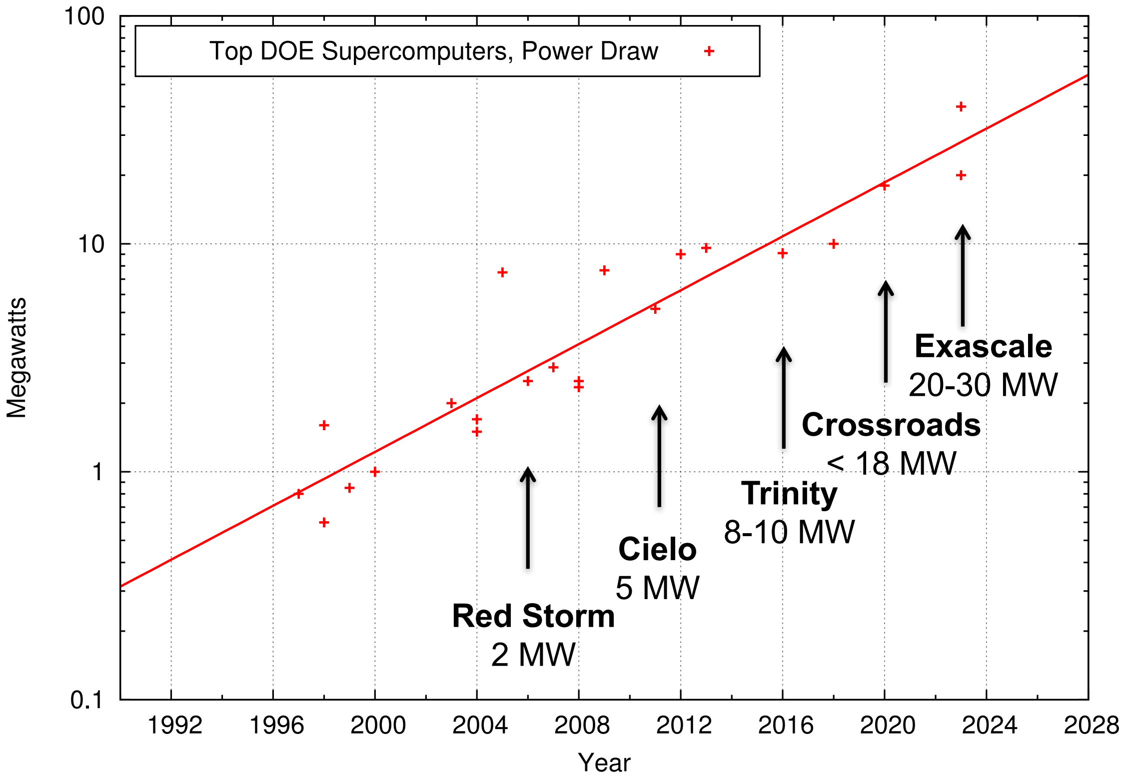
Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525.

Outline

- Background
- Trinity Advance Power Management (APM)
- Application profiling results
- Conclusions and path forward

Background

- Power usage of ATS-class systems increasing over time
  - Trinity is not power constrained, anticipate future systems will be
  - Investigating how to best use and operate future DOE platforms in a constrained power budget



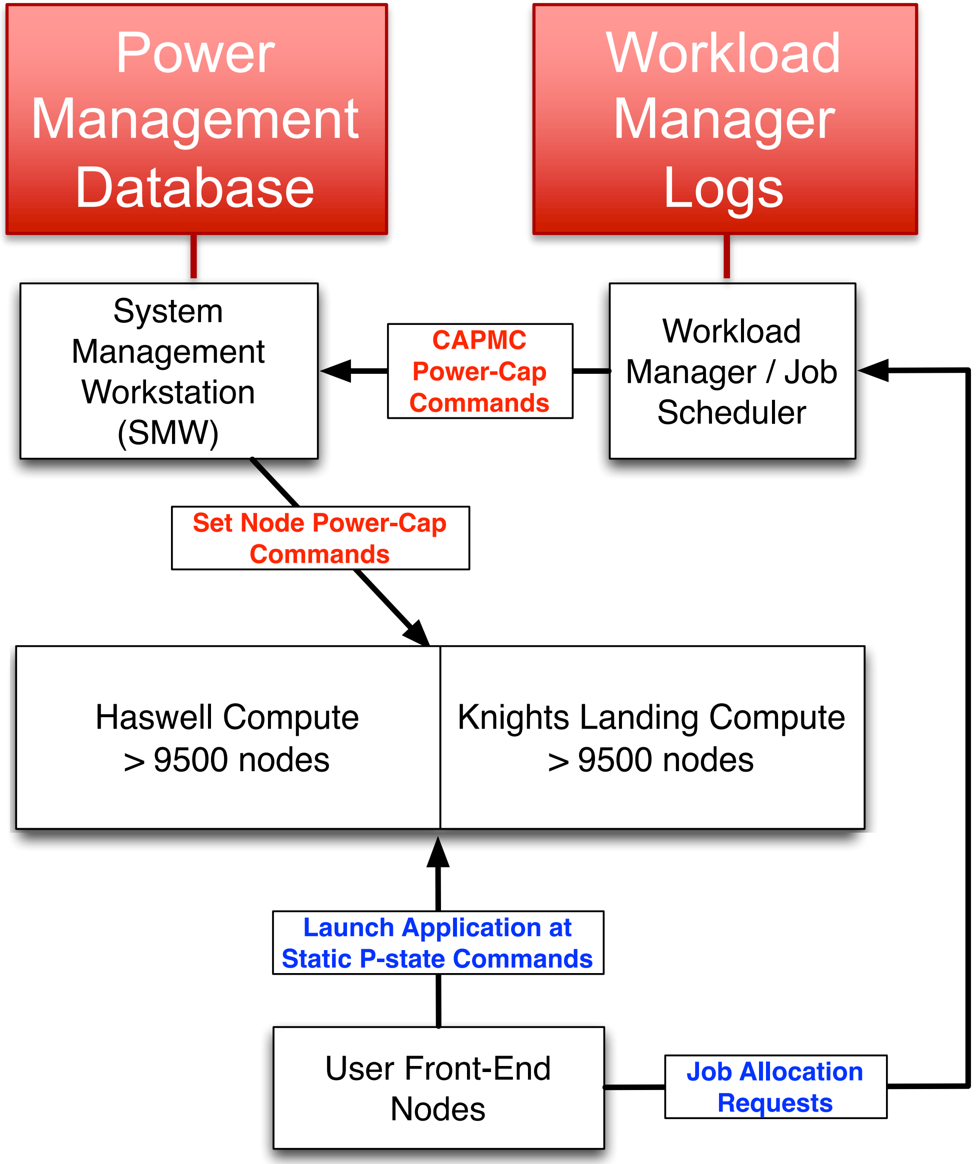
- Trinity Advanced Power Management Non-Recurring Engineering (APM NRE) Project
  - Cray – fundamental APM capabilities, Power API implementation
  - Adaptive – power-aware job scheduling and resource management

powerapi.sandia.gov

Example Use Cases

- A large job terminates early (because of finishing earlier than projected, being canceled or crashing) causing a significant drop in power usage, violating **system power floor** and **power ramp down contract terms with the local utility provider**
  - Equipment may fail
  - Contract violations may trigger financial penalties
- A few very large jobs are launched after a maintenance period, causing the system to significantly increase in power, first violating **power ramp up** contract terms then exceeding **system power ceiling**
  - Equipment may fail
  - Contract violations may trigger financial penalties
- For workloads that do not need to run at full power, allow **per app** or **per job power caps**
  - Reduce power usage of lower priority and low-CPU sensitive workloads, maybe wait less in queue
  - Allow re-allocation of power budget to higher value uses (e.g., a job that needs more power)
- Reporting back **power usage accounting** details to evaluate the full costs in ROI studies

Trinity Power Management Architecture



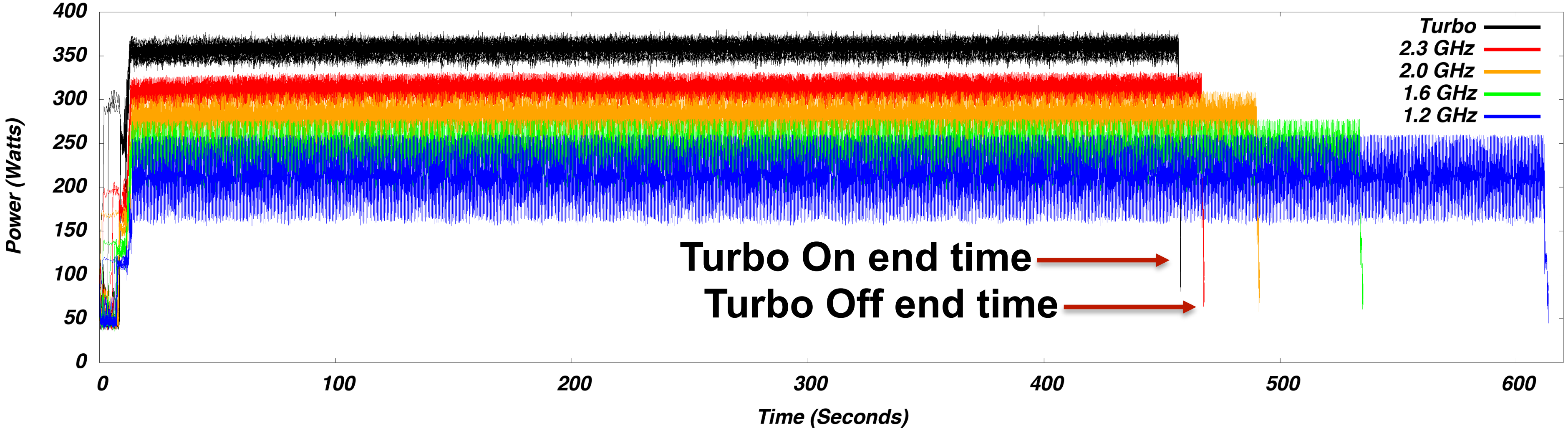
- A single management workstation controls system, the SMW
- Node-level power caps set from SMW, distributed to compute nodes via out-of-band management network
- Admins use `xtpmaction` command to set power caps manually
- Workload managers use Cray's CAPMC web API to set power caps + p-states
- Users may launch their job at a fixed p-state, default is P0 (turbo on)

Trinity APM Capabilities

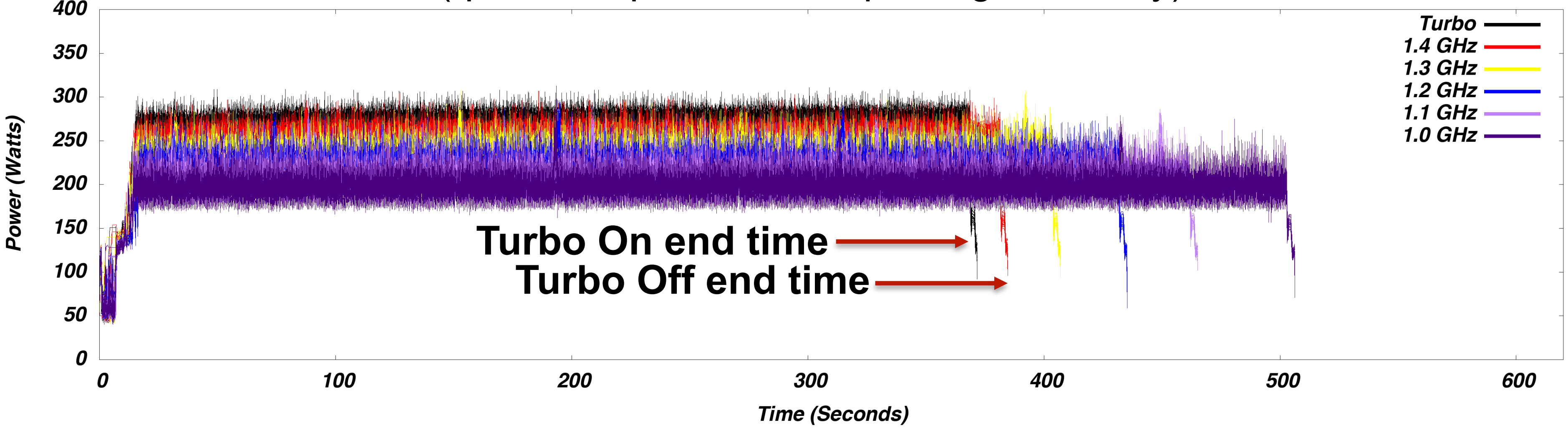
- System-level power ceiling and floor
  - Job scheduler only launches jobs that stay within ceiling limit
  - Floor implemented via c-state control, identified better options
- Job-level power ramp up management
  - Implemented in Torque prologue script, gradually increases power usage
- System-level power ramp down management
  - Implemented by gradually lowering c-state of idle nodes
- Job-level power templates
  - Users and admins can create power management templates and apply to jobs

SPARC Application Node-Level Power Profiles

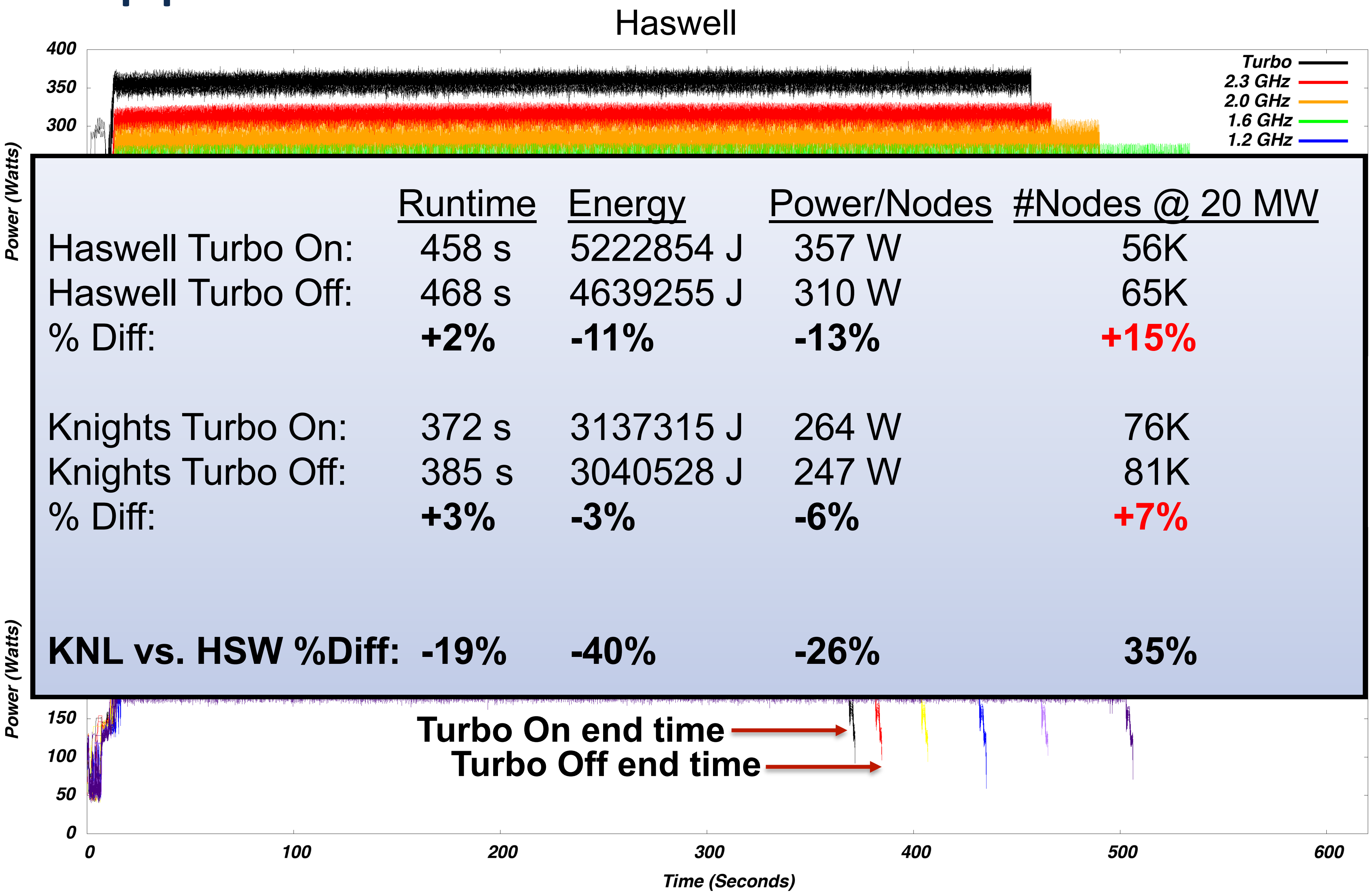
Haswell



Knights Landing
(quad-flat, pinned to on-package memory)

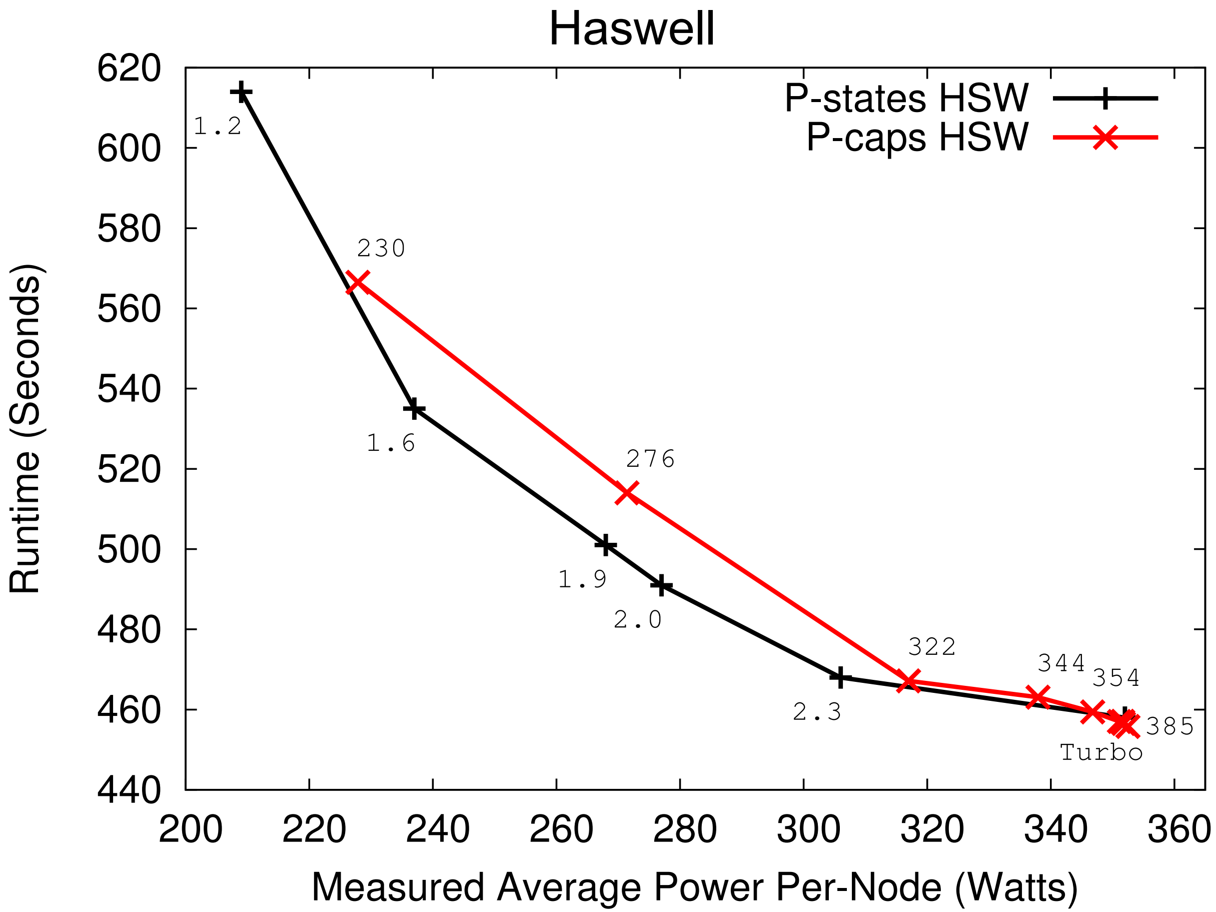


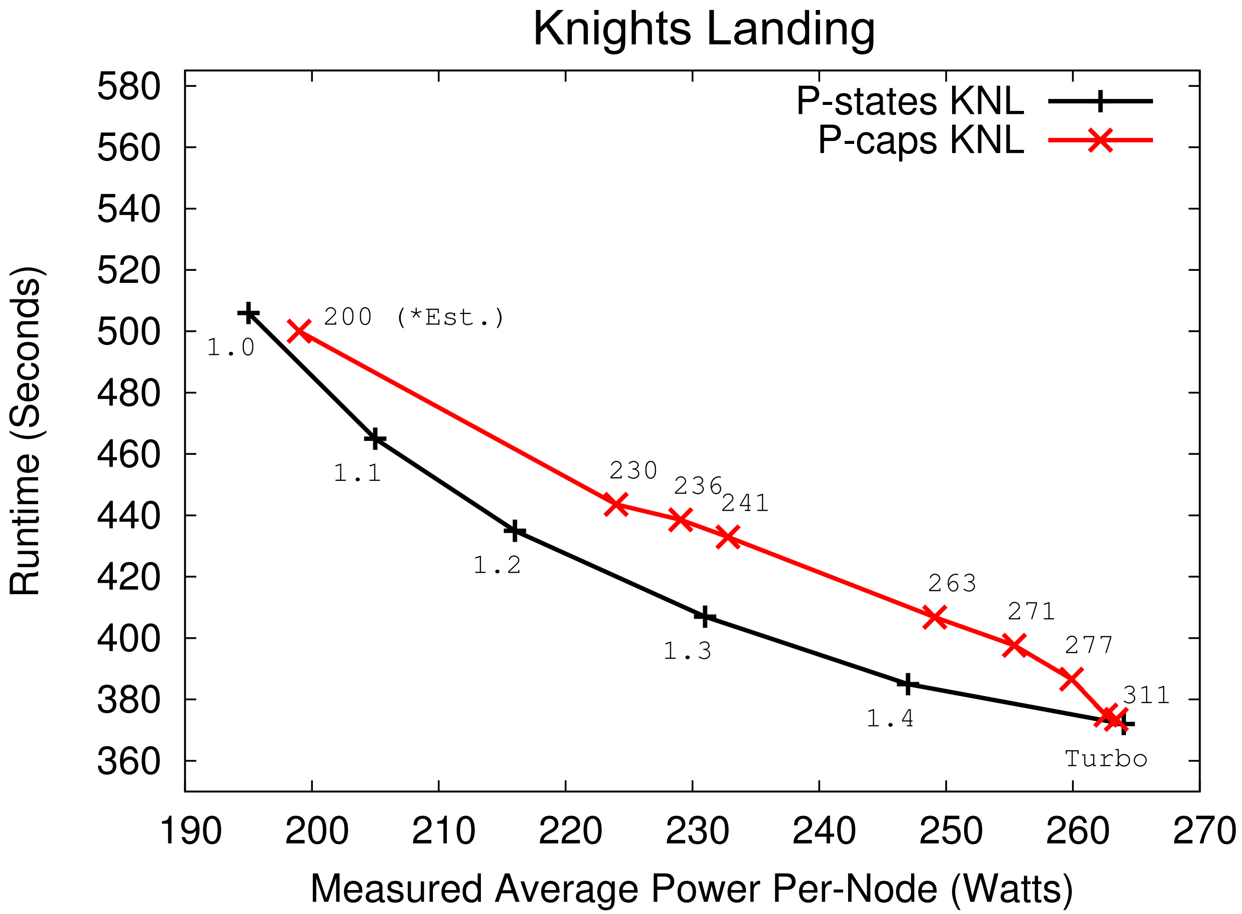
SPARC Application Node-Level Power Profiles



	Runtime	Energy	Power/Nodes	#Nodes @ 20 MW
Haswell Turbo On:	458 s	5222854 J	357 W	56K
Haswell Turbo Off:	468 s	4639255 J	310 W	65K
% Diff:	**+2%**	**-11%**	**-13%**	**+15%**
Knights Turbo On:	372 s	3137315 J	264 W	76K
Knights Turbo Off:	385 s	3040528 J	247 W	81K
% Diff:	**+3%**	**-3%**	**-6%**	**+7%**
KNL vs. HSW %Diff:	**-19%**	**-40%**	**-26%**	**35%**

Experimental Results





- SPARC – P-states control vs. Power Capping control
- Capping at the 75th percentile or above - similar performance, below that, performance degradation
- Performance constrained by frequently invoking the power capping mechanism

Next Steps

- Testing at scale with production workloads
- Enhance Power API to include
  - Power floor mechanism
  - Resource manager interface to indicate if a job is running or not on a given node
  - Automatic discovery of p-state to power usage correlations
- Consider implementing per-node power floor “burner” mechanism
  - Enables more precise control of power floor and ramp down rate
  - Created single node prototype, more plumbing needed to coordinate across nodes and interface with WLM

Conclusions

- Utilizing Trinity APM NRE capabilities to analyze DOE ASC workloads
- Developed and demonstrated power band and ramp rate management
  - Important for controlling system-level power usage
  - Identified challenges controlling power floor and ramp down; possible solutions
  - Implemented in MOAB/Torque workload manager, applicable to others
- Carrying forward Power API tools and analysis techniques to future DOE ASC platforms
  - Kokkos profiling interface power measurement plugins for PowerAPI
  - Tools for generating and analyzing point-in-time power plots
  - HPC power measurement taxonomy
    (IGSC'17: Evaluating Energy and Power Profiling Techniques for HPC Workloads)